

ロー・パワー・デザイン 特別企画 第2弾

Appendix 1

プロセッサの低消費電力設計

―― プロセッサの低消費電力化は，聖域なき総力戦

M.P.I

ここでは，いまや「低消費電力化技術の見本市」となりつつあるプロセッサ（マイクロプロセッサやマイクロコントローラ，ディジタル信号処理プロセッサなど）の技術動向を紹介する．全体的には，設計時にすべて決まってしまうような静的，固定的な対処法から，適応的，動的，アルゴリズミックな対処法へとシフトしているという．　（編集部）

本稿のタイトルは「プロセッサの低消費電力設計」となっていますが，プロセッサに特有の低消費電力化技術というものがあるわけではありません．プロセッサが論理LSIである以上，低消費電力化技術もまた，ほかの論理LSIの手法と共通しています．ただし，先端のプロセッサは熱的な限界に直面しているため，低消費電力化（というより，性能対電力効率比の向上）に対して，非常にシビアな要求があります．ほかのLSIと比べると，過激なまでの各種の低消費電力化手法が取り入れられている領域です．

● 実行時のリーク電流も大きな問題に

現在，適用されている低消費電力化の手法を整理すると，**図1**のようなセグメントに分けることができます．

一つの軸は，低減させたい対象がリーク電流による電力消費なのか，実際の機能動作のための電力消費なのか，というものです．ここで，「待機時消費電力＝リーク電流」というとらえかたでは，認識が不十分です．過去においてはこうした理解で十分だったのですが，微細化が進んだ先端プロセス，そしてそのようなプロセスで製造される最新のプロセッサでは，実行時のリーク電流もまた大きな問題となってきています．熱や電源の要求により，総消費電力の上限はある程度決まっています．こうした上限のある総消費電力から，リーク電流が多くの電力を盗み取ってしまうような状況に，すでに陥りつつあります．

リーク電流に電力を取られると，肝心の機能動作のために使える電力が減り，性能の低下をもたらすことになります．よってこの評価軸については，待機時や実行時といった単なる「時間的」なとらえかただけではなく，どのような経路で流れる電流への対策か，という「空間的」なとらえかたも加味しておいたほうがよいでしょう．

図1のカテゴリ分けのもう一つの軸は，設計時に決定される静的な手法か，実行時に動的に対処する手法かという点です．

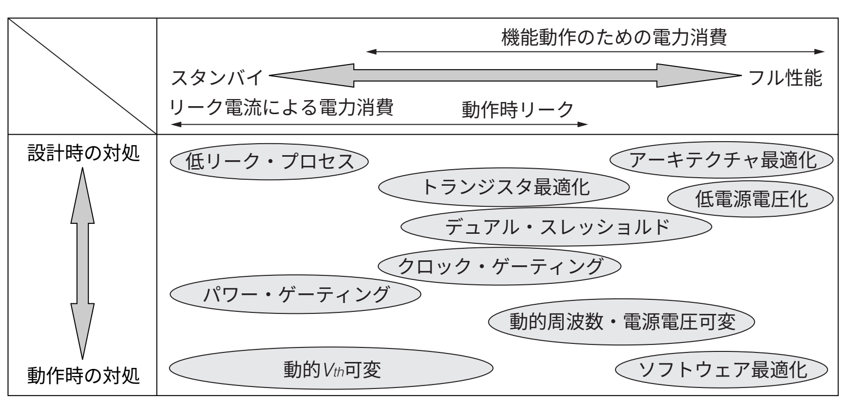
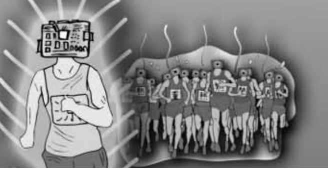


図1
代表的な低消費電力化手法
「低減させたい対象がリーク電流による電力消費なのか，実際の機能動作のための電力消費なのか」という軸と，「設計時に決定される静的な手法か，実行時に動的に対処する手法か」という軸で大ざっぱにカテゴリ分けした．

KeyWord　プロセッサ，リーク電流，クロック・ゲーティング，パワー・ゲーティング，LongRun2，SmartReflex，スレッショルド電圧，VTCMOS，MTCMOS，デュアル・スレッショルド，ゲート・リーク電流



これも細かく見ていけば，設計時以前に決まってしまうプロセス技術やデバイス技術によるものから，ソフトウェアに依存するものまで，さまざまな手法が含まれます．つまり，単純に2分できるものではありません．**図1**は，あくまでも大ざっぱな分類を示すものと考えていただければと思います．

● クロック停止と電源のしゃ断が対策の基本

プロセッサの場合，対象となるのはCMOS LSIなので，動的な消費電力を削減する場合，まず「クロックを止める」ということが基本中の基本です．設計時点で消費電力系のツールを駆使してクロック・ゲーティングを適用することが，普遍的な対処法となっています．この延長上には，クロック・ドメインの細分化なども含まれます．

これに対してリーク電流対策の基本としては，電源ドメインの細分化や電源しゃ断（パワー・ゲーティング），といった対処法が挙げられます．設計上の作業だけで比較的容易に対処できるクロック停止と比べると，電源しゃ断には対処すべき課題が多くなります．例えば，ハードウェアについては，異なる電源ドメイン（領域）間のアイソレーションやレベル・シフタなどが必要になります．

また，電源しゃ断からの復帰時に各フリップフロップのデータをどこからよみがえらせるのか，という問題もあります．いったん外部にデータを吸い出すという手法もありますが，現状では，フリップフロップを2重化し，低い電圧のバックアップ専用の電源ドメインに属する低リークのフリップフロップに保存する手法が主流となりそうです．もちろん，内部に電源しゃ断を行う回路を持たせるとなると，プロセス技術やデバイス技術による対処も必要になります．電源しゃ断の管理といった観点では，電力管理ソフトウェアの整備も重要な課題の一つです．

● x86プロセッサは電源電圧や動作周波数を動的に制御

消費電力の低減に対して2乗のオーダで効く電源電圧の低減は，プロセス技術の微細化の観点からも必然的な動向です．これをさらに効果的に利用する方策として，動的な電源電圧および動作周波数の可変技術があります．米国Transmeta社のLong Run技術（現在はLongRun2）が先鞭をつけ，米国Intel社や米国AMD（Advanced Micro Devices）社も追随しました．

動的な電圧制御を行ううえでは電源回路が重要であり，条件を追い込んで使うには，プロセスのばらつきや個体差まで考慮に入れる必要があります．このため，実回路上の温度センシングやプロセス/回路動作センシングなどを行う場合もあります．例えば米国National Semiconductor社は，ARM系のプロセッサに対して適応的な電圧制御技術を提供しています．また，米国Texas Instruments社もSmartReflexと呼ばれる動的電源制御技術を発表しています．

● スレッショルド電圧も可変に

困ったことに，電源電圧の低減には副作用があります．低電圧化が進むと，V_{th}（スレッショルド電圧）の低いトランジスタを採用することになり，これはリーク電流の増大という問題をもたらします．

これについては以前から，VTCMOS（variable threshold voltage CMOS）などの基板バイアスを使ったV_{th}制御技術や，すでに述べた電源しゃ断という面から，複数のV_{th}を使い分けるMTCMOS（multithreshold-voltage CMOS）技術などが対処法として挙がっていました．しかし最近では，待機時のリーク電流を制限するだけでなく，より動的に，動作時のV_{th}も可変にしようという流れになってきています．

複数のV_{th}を使い分けるという点では，設計の段階で，論理ブロック内の各パスのクリティカルさ（タイミングの厳しさ）を評価し，一つの論理ブロック内において異なるV_{th}のトランジスタからなるゲートを使い分けるデュアル・スレッショルド手法も一般化してきています（とくに，プロセッサ・コアを搭載するシステムLSIなどで適用されている）．もともとプロセッサ開発においては，トランジスタ・サイズを最適化するということは設計の基本だったのですが，それをV_{th}のレベルまで拡張したものととらえられます．

さらに微細化が進んでいるため，ゲート・リーク電流なども大きな問題となりつつあります．この面では，材料の検討を含む低リーク・プロセスの開発が第1の対処法ですが，パワー・ゲーティング技術の併用も必要になるでしょう．

● もっとも有効なのはパラレリズムの導入

これまで，おもに回路レベルやプロセス・レベルの低消費電力化技術について述べてきましたが，実は，プロセッサ，およびプロセッサ・コアを搭載するシステムLSIにおいていちばん効果の高い，そして典型的な低消費電力化の対策はアーキテクチャの最適化です．アーキテクチャ設計はこれまで，性能向上に対する有効性のみが強調されることが少なくありませんでした．しかし近年，低消費電力化のためのアーキテクチャ設計が注目を集めています．

基本は，パラレリズム（並列化）を駆使して，より低い周波数

で所望の性能を発揮させることです．これもプロセッサ特有の問題ではなく，多くの論理LSIに共通した低消費電力化の対策です．プロセッサの場合，スーパスカラ，マルチスレッド，マルチコアというぐあいに，歴史的に性能向上のためのパラレリズムの追求が盛んです．低消費電力化対策としてのパラレリズムについても，Intel社のCoreプロセッサなどですでに具現化されています．プロセッサ・コアを搭載するシステムLSIにおいて，低消費電力化のための各種アクセラレータの併用や超並列化などが進んでいるのは，この流れの延長と考えられます．

アーキテクチャ面だけでなく，ソフトウェア面での低消費電力化の手法も重要度を増しています．これは，前述のような各種の低消費電力化ハードウェアを制御する電力管理ソフトウェアを用意することだけではありません．低消費電力化のための一般のソフトウェアの最適化なども含みます．例えば，リソースの競合などにより多くのクロックや回路をむだに使ってしまう，といった状況そのものが問題とされるようになってきています．ソフトウェア設計者の負担は重くなりますが，最適な動作を行うソフトウェアの開発が望まれています．

● 非同期方式の導入も始まる

変わったところでは，非同期回路技術のプロセッサ設計への適用も始まっています．むだな部分を動かさないという意味で，非同期設計（昔ながらの「非同期」の概念とは手法的に異なるので，「セルフタイムド回路」と呼ばれることもある）は低消費電力化に向いています．実績は不明ですが，すでに英国ARM社がARM9系の非同期コアの商用提供を始めたようです．非同期系の技術には良い設計ツールがないことが長らくの課題でした．ARM社，そしてそれより以前からオランダPhilips Semiconductor社などは，オランダHandshake Solutions社のツールを用いてこの問題を解決しているそうです．

ほかにもプロセッサの低消費電力化技術には，チップ内相互接続，集積されるアナログ回路の最適設計，熱設計ともからむパッケージ設計，低電力化をねらう設計データのツール間互換性など，幅広い課題が存在します．

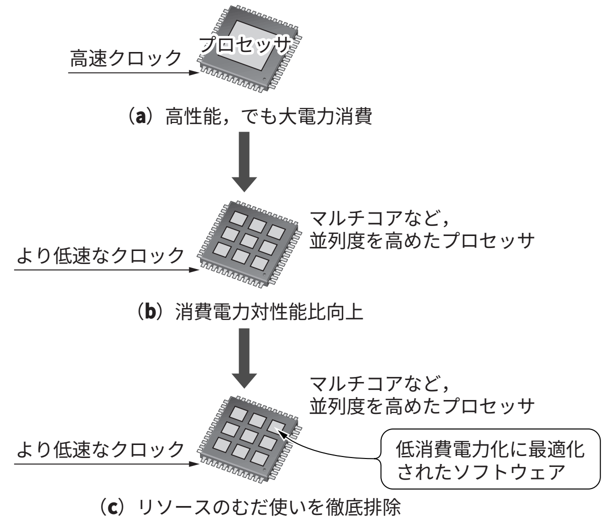


図2　プロセッサの低消費電力化のポイント
プロセッサの低消費電力化手法は，全体的には，設計時にすべて決まってしまうような静的，固定的な対処法から，適応的，動的，アルゴリズミックな対処法へとシフトしている．

全体的には，設計時にすべて決まってしまうような静的，固定的な対処法から，適応的，動的，アルゴリズミックな対処法へとシフトしています（**図2**）．もはや，LSIの消費電力は単一のファクタで一意に決められるようなレベルにはありません．前述してきたように関連するファクタが非常に多いうえ，相互に複雑にからみあっています．状況に応じて空間的にも時間的にも最適な制御を行っていかないと，低消費電力化を達成できなくなりつつあります．

プロセッサの低消費電力設計は，今や総力戦なのです．

M.P.I
マイコン開発マネージャ